

CC4302 Sistemas Operativos – Tarea 2 – Semestre Primavera 2023

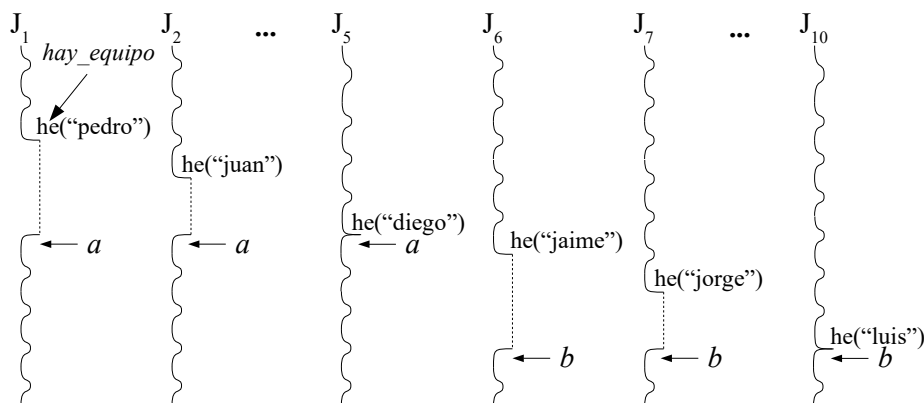
Se necesita formar equipos de 5 jugadores de baby-fútbol. Los jugadores son representados por threads que invocan la función *hay_equipo* indicando como argumento su nombre. Esta función espera hasta que 5 jugadores hayan invocado la misma función retornando un arreglo de 5 strings con los nombres del equipo completo. Este es un ejemplo del código de un jugador:

```
void *jugador(void *ptr) {
    char *nombre= ptr;
    for (;;) {
        dormir();
        char **equipo= hay_equipo(nombre);
        jugar_baby(equipo);
        beber_cerveza();
    }
}
```

Programa la función *hay_equipo*. El encabezado es el siguiente:

```
char **hay_equipo(char *nombre);
```

Las funciones *dormir*, *jugar_futbolito* y *beber_cerveza* son dadas. El siguiente diagrama muestra un ejemplo de ejecución:



Observe que la llamada a *hay_equipo* espera hasta que se haya formado un equipo con 5 jugadores. Los primeros 5 jugadores (J₁ a J₅) forman el equipo *a* y por lo tanto sus llamadas a *hay_equipo* retornan el mismo arreglo *a* con los 5 nombres del equipo: "pedro", "juan", ..., "diego". Los siguientes 5 jugadores (J₆ a J₁₀) forman el equipo *b* y sus llamadas a *hay_equipo* retornan el arreglo *b*, distinto de *a*, con los nombres "jaime", "jorge", ..., "luis". Resuelva el problema usando variables globales como un mutex, una condición, etc.

Si necesita inicializar variables globales o liberar recursos, hágalo en las funciones *init_equipo* y *end_equipo*.

Restricciones: Para programar las funciones solicitadas Ud. debe usar un mutex y una sola condición. No está permitido usar múltiples condiciones.

Instrucciones

Baje *t2.zip* de U-cursos y descomprímalo. El directorio *T2* contiene los archivos *test-equipo.c*, *Makefile*, *equipo.h* (con los encabezados requeridos) y otros archivos. Ud. debe programar en el archivo *equipo.c* las funciones solicitadas. Defina otras funciones si las necesita.

Pruebe su tarea bajo Debian 11 de 64 bits. Ejecute el comando *make* sin parámetros. Le mostrará las opciones que tiene para compilar su tarea. Estos son los requerimientos para aprobar su tarea:

- *make run* debe felicitarlo por aprobar este modo de ejecución.
- *make run-g* debe felicitarlo.
- *make run-san* debe felicitarlo y no reportar ningún incidente en el manejo de memoria.
- *make run-thr* debe felicitarlo y no reportar ningún datarace.

Cuando pruebe su tarea con *make run* en su computador asegúrese de que esté configurado en modo alto rendimiento y que no estén corriendo otros procesos intensivos en uso de CPU al mismo tiempo.

Invoque el comando *make zip* para ejecutar todos los tests y generar un archivo *equipo.zip* que contiene *equipo.c*, con su solución, y *resultados.txt*, con la salida de *make run*, *make run-g*, *make run-san* y *make run-thr*.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *equipo.zip* generado por *make zip*. Recuerde descargar el archivo que subió, descargar nuevamente los archivos adjuntos y volver a probar la tarea tal cual como la subió a U-cursos. Solo así estará seguro de no haber entregado archivos incorrectos. Se descuenta medio punto por día de atraso. No se consideran los días de receso, sábados, domingos o festivos.